

コア温度情報を用いた OS による 細粒度パワーゲーティング制御方式の設計

木村 一樹^{†1} 近藤 正章^{†4} 天野 英晴^{†3}
宇佐美 公良^{†5} 中村 宏^{†2}
佐藤 未来子^{†1} 並木 美太郎^{†1}

演算ユニットにランタイムパワーゲーティング (PG) 技術を施した省電力プロセッサコア Geyser において、OS により PG 動作を制御することでハードウェア単体での PG より電力削減効果を高めることが期待されている。本研究では、PG 実施時のオーバーヘッドと電力削減量の損益分岐点がコアの温度により変化することに着目し、ランタイムにコアの温度情報を取得して OS により PG 実施ポリシーの決定を行う制御方式を提案する。またこの方式について FPGA ボード上に構築した評価システムに実装し評価を行い、演算ユニット全体の平均リーク電力を最大約 11%削減した。

Design of OS Based on Fine Grain Power Gating Control By Temperature Information of CPU Core

KAZUKI KIMURA,^{†1} MASAOKI KONDO,^{†4}
HIDEHARU AMANO,^{†3} KIMIYOSHI USAMI,^{†5}
HIROSHI NAKAMURA,^{†2} MIKIKO SATO^{†1}
and MITARO NAMIKI^{†1}

‘Geyser’, the processor core with a fine grain power gating technique, requires further power savings by controlling the behavior of power gating with OS support. This paper describes an approach to improve power saving efficiency of Geyser that is the method by runtime temperature information of CPU core to decide the behavior of power gating, focusing on the fact that the break-even point of power saving efficiency varies with the temperature. The leakage power consumption of 4 computing units has been reduced by up to 11% with the method on the FPGA evaluation board.

1. はじめに

現代の高度情報化社会を支えるシステム LSI は、動作周波数の高速化と製造プロセスの微細化により著しい性能向上を達成してきた。しかし動作周波数の高速化に従い LSI を構成するトランジスタを駆動するためのダイナミック電力の増大と、それに伴う LSI の発熱が問題となった。一方、プロセスの微細化技術は LSI の動作電圧の低減をもたらす、ダイナミック電力の増大を抑制することに寄与した。ところが、プロセスの微細化は同時に漏れ電流を増大させリーク電力を顕在化させる要因となった。近年では LSI の消費電力のうちリーク電力の占める割合が支配的となってきており、これを抑える技術の確立が求められている。

LSI の消費電力の問題に対しては、ハードウェア、ソフトウェアの各分野で様々な取り組みが行われてきた。半導体レベルでは、ボディバイアスや VDD コントロールなどの技術が、回路レベルでは機能ブロックごとのクロックゲーティングやパワーゲーティングなどの技術が提案された。またソフトウェアにおいてもメモリ管理機構やスケジューリングの工夫によって消費電力の削減を図る取り組みが行われている⁴⁾ ほか、ソフトウェアとハードウェアの連携によって省電力化を実現する研究も行われている。

例えば Donald²⁾ らは、マルチコアプロセッサにおけるコアの製造バラつきに着目し、消費電力の少ないコアから優先的にタスク割り当てを行う技術を提案した。また Ramamurthy³⁾ らは、性能を考慮しつつメモリ配置を工夫し、メモリデバイスの電源管理を行う技術を提案した。これら既存の省電力研究においては、処理性能と消費電力のトレードオフが課題とされた。

これに対し、科学技術振興機構 (JST) の戦略的創造研究推進事業「CREST」における「情報システムの超低消費電力化を目指した技術革新と統合化技術」¹⁾ の一プロジェクトで

^{†1} 東京農工大学
Tokyo University of Agriculture and Technology

^{†2} 東京大学
The University of Tokyo

^{†3} 慶應義塾大学
Keio University

^{†4} 電気通信大学
The University of Electro-Communications

^{†5} 芝浦工業大学
Shibaura Institute of Technology

は、演算ユニットごとに命令サイクル単位でパワーゲーティングを施し、性能を劣化させずにリーク電力の削減を図る技術を実施した MIPS R3000 アーキテクチャベースのプロセッサコア、Geyser を試作した。またこのプロジェクトにおいて東京農工大学並木研究室では、Geyser 向けに組込み OS (Geyser OS) を移植し、OS レベルからのパワーゲーティング制御について研究を行っている。

本稿では、OS によるパワーゲーティング動作の制御の一方式として、コア温度情報を用いる方法を提案する。この方式は、パワーゲーティングに伴うオーバーヘッドと電力削減量の損益分岐点が温度により変化するという特性に着目したものである。また、実際に提案方式を OS に組み込み、FPGA ボード上に構築した Geyser の評価環境において電力削減効果の検証を行った結果を示す。

2. Geyser 概要

本章では省電力プロセッサコア Geyser の概要と、その省電力効果について述べる。

2.1 Geyser のアーキテクチャとパワーゲーティング

Geyser⁵⁾ は、MIPS R3000 をベースとしたプロセッサコアである。コア内に演算ユニットである ALU, SHIFT, MULT, DIV が存在し、それぞれの演算ユニットに対して命令サイクルごとの細かい粒度で動的にパワーゲーティングを行うことができる。Geyser の試作チップである Geyser-0 における評価では、パワーゲーティングの効果によりコア全体で平均約 47% の電力削減効果を確認している⁵⁾。

2.1.1 細粒度パワーゲーティングの制御インタフェース

Geyser では、パワーゲーティングの粒度をソフトウェア側から制御するために、特権レジスタである PGStatus レジスタを有し、パワーゲーティング対象の各演算ユニットに対し、次の三つのスリープポリシーを定めることができる。

- 動的パワーゲーティング
- キャッシュミス時のみスリープ
- 常にアクティブ (スリープしない)

OS は PGStatus レジスタを通して、細粒度パワーゲーティングのスリープポリシーを制御することが可能である。Geyser では「動的パワーゲーティング」が標準のポリシーであるが、期間の短いスリープが頻発すると、オーバーヘッドにより消費電力が増えてしまうことがある。この場合に「キャッシュミス時のみスリープ」や「常にアクティブ (スリープしない)」のポリシーを設定することで省電力効果の劣化を防ぐことが期待できる。

3. 課題と目標

本章では、先行研究におけるスリープポリシー制御方式の課題と、本研究の目標を示す。

3.1 先行研究の課題

先行研究⁷⁾ で提案した OS による省電力機構は、パワーゲーティングによる演算ユニットのスリープサイクルが電力的な損益分岐点を超えていたかどうかを入力情報とする。これを取得するために、パワーゲーティング実施時の演算ユニットの消費電力を推算するためのハードウェアモジュールである PG パフォーマンスカウンタを使用したが、このモジュールはスリープサイクル記憶のため、パワーゲーティング対象の演算ユニットに比べ回路規模が大きい。つまり実際の運用を考えると、このモジュールを使用しない制御方式でなければ演算ユニットのパワーゲーティングによる消費電力削減効果を得ることは難しいといえる。

また、先行研究での評価は RTL シミュレーションにより行った。RTL シミュレーションでは主記憶をレジスタアレイにより実現しているが、アクセスコストが少なくキャッシュなどより高速な記憶装置と差別化されないため、評価環境として実計算機との差異が大きい。I/O や各種割込みなどの動作も含め、より実計算機に近い評価環境での評価が求められる。加えて、評価についてはコア温度が一定の場合のみに対して行われており、温度が変化する場合については想定されていなかった。

3.2 本研究の目標

本研究では、PG パフォーマンスカウンタを使用せずにコアの温度情報を用いるスリープポリシー制御方式を提案する。この方式では温度が変化する場合を想定し、パワーゲーティングによる電力的損益分岐点が温度により変化することに着目して OS が動作時のコア温度から最適なスリープポリシーを設定する。この方式について、ソフトウェアによるスリープポリシー制御が行われない場合と比べて

- 演算ユニットの平均リーク電力
- 電力的損益分岐点に満たないスリープの割合

が削減されることを確かめる。また制御機構の処理によるプログラムのオーバーヘッドも評価し、これが大きくなりすぎないことを目指す。これらの評価は、前述の理由により、FPGA ボード上に構築した Geyser の計算機環境において行う。また、実際に温度の変化をエミュレーションし、評価する。

4. 細粒度パワーゲーティングの電力評価モデル

本章では, Geysler アーキテクチャにおける細粒度パワーゲーティングの電力の特徴について説明し, 提案方式の評価指標である電力的損益分岐点に満たないスリープの割合, BEP(Break Even Point) ミス率について定義する.

4.1 細粒度パワーゲーティングによる電力遷移

Geysler における細粒度パワーゲーティングでは, ユニットのアクティブ状態とスリープ状態の遷移の際に電力オーバーヘッドが生じる. このオーバーヘッドについて, 図 1 に模式的な電力グラフを示す.

グラフを構成する各領域は, 以下の消費電力を示している.

- 領域 A: アクティブ時リーク電力
- 領域 B: アクティブ スリープ遷移時のオーバーヘッドリーク電力
- 領域 C: スリープ時リーク電力 (過渡現象による曲線)
- 領域 D: スリープ アクティブ遷移時のオーバーヘッドリーク電力

領域 B と D で示される電力がパワーゲーティングによるオーバーヘッドであり, これを上回るだけの電力を削減するのに十分な長さのスリープでなければ, 省電力効果は得られない. 次に, このオーバーヘッドを考慮した損益分岐点 (BEP: Break Even Point) を定義する.

4.2 BEP 及び BEP ミス率の定義

常にアクティブ状態であった場合に消費する電力よりも状態遷移によるオーバーヘッドを考慮してそれでもスリープ状態へ移行した方が, 消費電力が削減できる点 (サイクル数) を電力損益分岐点 (BEP) と呼ぶ. BEP は, 温度と演算ユニットの特性によって異なり, ある演算ユニットにおいてスリープ期間を t , 温度を T とした時, スリープ遷移におけるエネルギーオーバーヘッドを $E_{sleepOH}(T)$, スリープから動作状態に遷移する際のエネルギーオーバーヘッドを $E_{wakeUpOH}(T)$, スリープすることによって削減されるエネルギー $E_{sleep}(t, T)$ とすると, スリープと動作状態の遷移において,

$$E_{sleep}(t, T) < E_{sleepOH}(T) + E_{wakeUpOH}(T)$$

を満たすときの, スリープ期間 t を $BEP(t_{bep})$ と定義する. t_{bep} を越えるスリープ期間を得られないときは, スリープしないと比べて消費エネルギーは増加する. また BEP は, 演算ユニットごとに異なり, さらに演算ユニットの温度によって変化する. Geysler では試作した実チップによる測定から, 表 1 に示す, BEP が与えられている. ここで BEP ミスを, 細粒度パワーゲーティングによって発生するオーバーヘッドがスリープによって削減された電

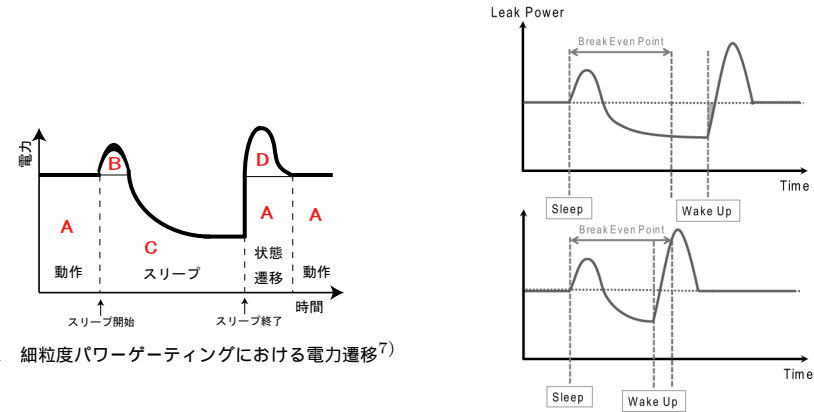


図 1 細粒度パワーゲーティングにおける電力遷移⁷⁾

図 2 BEP の定義 (上: BEP を満たす PG 下: BEP ミスする PG)⁷⁾

表 1 細粒度パワーゲーティングのブレイクイーブンポイント [単位: サイクル]⁵⁾

	25	65	100	125
ALU	74	26	12	8
SHIFT	114	38	16	10
MULT	74	22	10	8
DIV	44	14	6	2

力を上回ってしまった場合, つまり細粒度パワーゲーティングを行うことによって逆に電力ロスが生じる状況として定義する. 図 2 に, BEP を満たす状況と BEP ミスする状況を図示する.

図 2 の上のグラフは, WakeUp のタイミングが BEP を超えており, 電力削減を実現していることを示す. 一方で下のグラフは, WakeUp が BEP より早いため, 十分に電力を削減できず電力ロスを生じている場合をあらわしている.

BEP ミスが生じる頻度の評価指標として, BEP ミス率を定義する. ある評価期間におけるスリープサイクルを $t_{sleepAll}$, 長さ i サイクルのスリープを t_i とすると, BEP ミス率は次式で表される.

$$T = \{t_i | i < bep\}$$

$$BEP_{MissRate} = \frac{\sum_{t_i \in T} t_i}{t_{sleepAll}} \quad (1)$$

この式は、BEP ミス率が高ければ高いほどパワーゲーティングにおける電力のロスが大きくなることを意味する。

4.3 電力推算モデル

RTL シミュレーションや FPGA による評価環境では、パワーゲーティングの効果を電力として実測することができない。そこで、演算ユニットの電力をモデル化した上で、これにパワーゲーティングによるスリープ頻度の実測結果を適用することで、演算ユニットの電力を推算するという方法を取る。

まず、Geysler の演算ユニットの電力をアクティブ時とスリープ時に分けて考える。アクティブ時の電力は、さらに動作に伴うスイッチング動作により消費されるダイナミック電力と、定常的に消費されるリーク電力とに分けられる。一方スリープ時の電力は、ユニットが動作状態にないため、ダイナミック電力の消費は無視することができリーク電力のみとして考えられるが、先述したようにスイッチングに伴うオーバヘッドが存在し、またリーク電力は過渡的に遷移する。このことを考慮に入れた上で、スリープ時の平均リーク電力をスリープの長さ（スリープサイクル数）に対して変化する値として扱う。この相関係係について先行研究により演算ユニットごとに実験値が得られている。この実験値を、プログラム実行時のスリープ頻度をスリープサイクル数ごとに記録したデータに適用することで、全スリープ領域における平均リーク電力を推算することができる。演算ユニットが N サイクルスリープしたときの平均リーク電力を P_{sN} とすると、そのユニットの全スリープ期間の平均リーク電力 P_s は、 N サイクルのスリープ頻度 M_N と総スリープサイクル数が T_{sleep} より

$$P_s = \sum_N (P_{sN} \times \frac{M_N \times i}{T_{sleep}}) [W] \quad (2)$$

で与えられる。

本稿における電力評価では、評価対象の制御方式に対してスリープ時平均リーク電力の比較によって評価を行う。

5. コア温度情報を用いた制御方式の設計

本章では、提案方式の設計について述べる。

5.1 提案方式の概要

本研究では、コアの温度情報を用いる手法を提案する。先述した BEP は温度に相関する

値で、低いほど短く、温度が高いほど長くなるという性質を持つ。この性質に着目し、温度情報を入力とすることで単純ながらも効果の高い PG 制御を実現する。またオンチップ温度センサは多くのプロセッサに搭載されている汎用的なモジュールであり、プログラムからも簡単に温度を取得できるので、実用的であるといえる。

5.2 PG ポリシーの決定方法

Geysler の通常の PG ポリシーである「動的パワーゲーティング」では演算ユニットを使用しない場合に積極的にスリープするため、スリープサイクル数の分布が広範なものとなりやすい。これにより BEP より短いサイクル数のスリープが高頻度に現れると省電力効果が得られなくなる。提案方式ではそのような傾向を持つスリープによる消費電力の削減効果の劣化を防ぐため、温度により動的に PG ポリシーを切り替える。特に BEP がキャッシュミスサイクル数の下限値付近にある場合は、「キャッシュミス時のみスリープ」のポリシーとすることでそれよりも短いサイクル数のスリープによる電力ロスを抑えられる。

そこで、次のような方式を考える。

(1) 2 ポリシー方式

スリープサイクル数のうちキャッシュミス発生時のスリープよりも短いスリープサイクル数の領域から 1 つの値を決め、この値と BEP グラフの交点から下ろした垂線と横軸との交点にあたる温度を閾値 1 (T_{TH1}) とし、コア温度 T_{core} がこの T_{TH1} より高い場合には「動的パワーゲーティング」ポリシーを、低い場合には「キャッシュミス時のみスリープ」のポリシーをそれぞれ使用する (図 3)。

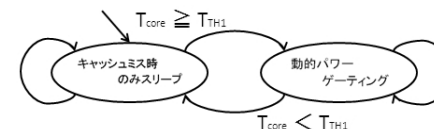


図 3 2 ポリシー方式

(2) 3 ポリシー方式

前述の「2 ポリシー方式」に加え、 T_{TH1} よりも高い何らかの温度を閾値 2 (T_{TH2}) とし、 T_{TH1} と T_{TH2} の間では「キャッシュミス時のみスリープ」、 T_{TH2} 以上では「常にアクティブ」とする (図 4)。このとき、閾値 2 (T_{TH2}) は BEP がキャッシュミス時のスリープサイクル数の加減よりも大きい値となるように設定する。

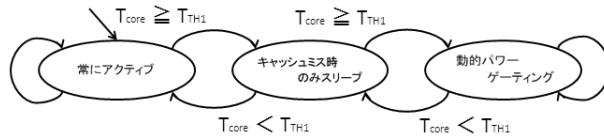


図 4 3 ポリシー方式

表 2 Geyser on FPGA 仕様

FPGA ボード	Xilinx ML501 (VIRTEX5)
Geyser コア周波数	40MHz
キャッシュ	命令 8KB , データ 8KB
主記憶	DDR 256MB, SRAM 1MB

6. 実装

本章では提案方式の OS への実装などを記す。

6.1 Geyser OS への組込み

Geyser OS とは、東京農工大学並木研究室で開発している組込み向け OS であり、Geyser アーキテクチャ上で動作するようにポーティングが行われている。Geyser OS はタスク管理、仮想メモリ管理、例外・割込み管理などの基本的な機能から構成され、ユーザプログラム側に対してはシステムコールを提供している。

PG ポリシー制御の動作タイミングの最適値は未検証であり、先行研究⁷⁾ ではタスクスケジューラと同期したタイミングが採用されていた。本研究ではタスクスケジューラとは独立した動作タイミングで提案方式が動作できるよう、タスクスケジューラ割込みを発生させるものとは別のタイマを使用して PG ポリシー制御を起動する。このため PG ポリシー制御プログラムに Geyser OS の割込みベクタテーブルを独自に割り当てた。PG ポリシー制御プログラムを起動するタイマの周期について、本研究では最適値の検証を省略するが、実験値として 100[ms] に設定した。この周期で PG ポリシー制御が起動すると、温度エミュレーションプログラムを通してそのときの温度を取得し、その温度と閾値を比較して PG ポリシーを決定し PGStatus レジスタに書き込んで、割込みから復帰する、という動作を行う。

6.2 FPGA ボードを用いた評価基盤

本研究では、評価環境として FPGA ボード上に構築した Geyser による計算機環境、Geyser on FPGA を用いた。Geyser on FPGA は主記憶として SDRAM などを持つほか、タイマや UART、経過サイクル数を計測するためのサイクルカウンタ（アップカウンタ）などの各種入出力がある。

またパワーゲーティングによる電力削減効果を推算するためのスリープ頻度の計測ツールとして、PG パフォーマンスカウンタが実装されている。先行研究ではこのモジュールによる計測結果を参照して制御を行ったが、本研究の制御方式はこのモジュールを参照しない。

制御結果の評価に限り、OS でこのパフォーマンスカウンタに記録したスリープ頻度情報を読み出して、BEP ミス率や平均リーク電力の推算を行う。

6.3 コア温度エミュレーション

コア温度についても、FPGA ボードでは実物の LSI に即した温度を実測することはできないため、サイクル数をパラメータとして一定の値域において線形及び周期的に変化するいくつかの温度テストケースをソフトウェアでエミュレーションする。このエミュレーションプログラムは PG ポリシー制御プログラムからシステムの経過サイクル数を受け取って時間経過に換算し、あらかじめ設定したテストケースに則った「現在のコア温度」をエミュレートする。このプログラムはパフォーマンスオーバヘッドの評価対象には含めない。

温度は 25 から 125 までの範囲をエミュレーションでき、単精度浮動小数点数として読み出すことができる。評価プログラムでは、この温度情報をパラメータとして、BEP や平均リーク電力を実験値から多項式補間により内挿して用いる。

7. 評価と考察

本稿における提案方式のうち、「2 ポリシー方式」について実際に評価を行った。Geyser OS 上の 2 つのタスクでそれぞれ評価プログラムとして QSORT, MATRIX, Dhrystone を実行した結果を示す。温度は、実行開始とともに 25 から線形に上昇し、30 秒で 65 に到達するケースをエミュレーションした。本章で示す BEP ミス率とスリープ時平均リーク電力は、温度が変化している区間に対する評価結果である。評価プログラムはタスク内で繰り返し実行され、評価が終了するまで動作し続ける。

PG ポリシーを変更する閾値温度については、次に挙げる二種類のサイクル長に基づいて決定し、評価に用いた。それぞれの閾値温度は表 3 のとおりとなる。

(1) キャッシュミスによるスリープ長の下限值付近

Geyser on FPGA では、キャッシュミスによるメモリアクセスコストが概ね 60 サイクルから 80 サイクルの間の値となる。このうち下限値の 60 サイクルよりも BEP が長くなる場合、つまりキャッシュミスによるスリープより短いスリープが、いずれも

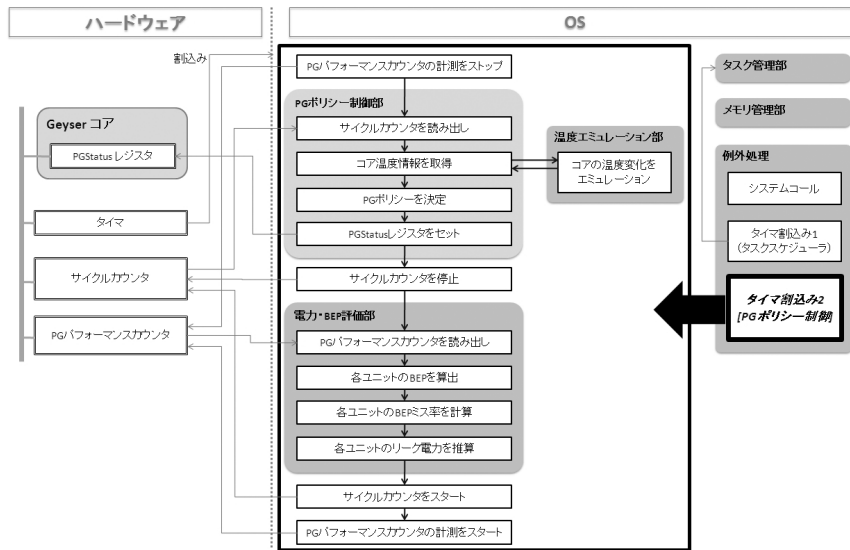


図5 PGポリシー制御部と評価部の構成図

BEP ミスとならないような場合のみ動的 PG ポリシーとする。

(2) キャッシュミスによるスリープ長の下限值未満の領域の重心

前述の下限值よりも短いスリープについて、度数分布を一様と仮定したときの重心となるサイクル数に閾値を設定する。このケースにおいて動的 PG ポリシーとなった場合、スリープのうち一部は BEP ミスとなるが、BEP ミスとならない領域のスリープがそれを上回る可能性が高いと考えられ、したがってトータルで BEP ミスを削減し省電力効果が期待できる。このサイクル数は、計算よりおよそ 43 サイクルとなる。

表3 ポリシー変更の閾値温度 []

	ALU	SHIFT	MULT	DIV
制御 (1) の閾値	34	49	33	25
制御 (2) の閾値	47	61	44	26

また比較対象として提案方式の制御を行わず動的 PG ポリシーに固定したケースと、パ

ワーゲーティング自体を行わなかったケースを扱う。それぞれについて、以降の評価では次の語を用いて説明する。

- 制御なし: スリープポリシーに対する操作を行わず、常に「動的 PG」ポリシーでパワーゲーティングが行われている状態。
- 制御あり (1): 前述の「キャッシュミスによるスリープ長の下限值付近」に基づく温度を閾値とした、提案方式による制御。
- 制御あり (2): 前述の「キャッシュミスによるスリープ長の下限值未満の領域の重心」に基づく温度を閾値とした、提案方式による制御。
- PG なし: 細粒度パワーゲーティングを行わないケース。

7.1 BEP ミス率

ALU, SHIFT ユニット, MULT ユニットの BEP ミス率を、それぞれ図 6, 7, 8 に示す。ALU では、いずれの評価プログラムに対しても BEP ミス率を改善することができた。制御 (1) ではおよそ 13%, 制御 (2) ではおよそ 47% の削減となった。一方 SHIFT ユニットでは、制御を加えず常に動的パワーゲーティングを行った場合に比べて、BEP ミス率が増える結果となった。

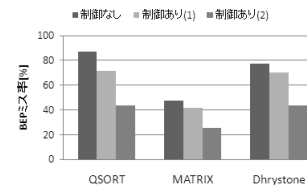


図6 ALU の BEP ミス率

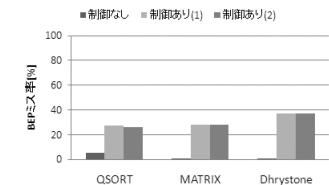


図7 SHIFT ユニットの BEP ミス率

MULT ユニットでは、もともと BEP ミス率が小さく、PG ポリシー制御による大きな変化は見られなかった。なお DIV ユニットは評価プログラムでは使用されず、常にスリープしていた。このため BEP ミス率は 0 であり、グラフの掲載は省略する。

提案方式による PG ポリシー制御の評価の結果、演算ユニットごとに効果の違いが現れた。このことから、各演算ユニットのスリープ頻度の特性が BEP ミス率の変化に大きく影響していると考えられる。ALU では BEP ミス率が減少したが、これは ALU でのスリープが短いサイクルで頻発しており、コア温度が低く BEP が長い領域において動的パワーゲーティングを制限することで BEP ミスとなるようなスリープを効果的に抑制したからである。

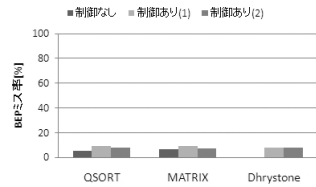


図 8 MULT ユニットの BEP ミス率

う．反対に SHIFT ユニットでは BEP ミス率が増える結果となった．これは SHIFT ユニットにおけるスリープが ALU のように短いサイクルに集中しておらず，キャッシュミス時のスリープ長を超え数百サイクルに至るようなものが多く含まれているため，動的パワーゲーティングの制限が返って効果的なスリープも抑制してしまったものと考えられる．DIV ユニットと MULT ユニットについては，使用頻度が低いいため動的パワーゲーティングを行っても BEP ミスとなるような短いスリープが発生することが少なく，PG ポリシー制御の効果が発揮されなかったと考えることができる．

7.2 消費リーク電力

図 9 から図 12 に，それぞれ ALU，SHIFT，MULT，DIV の各演算ユニットのスリープ時消費リーク電力を示す．ALU ではタスクの種類によらず，制御あり (1) では平均で約 14%，制御あり (2) では平均約 39% リーク電力が削減された．一方 SHIFT ユニットでは，制御あり (1) で約 40%，制御あり (2) では約 55%，リーク電力が増大した．

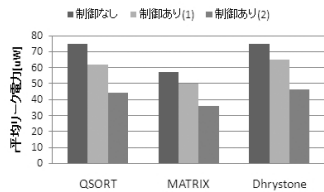


図 9 ALU のスリープ時平均リーク電力

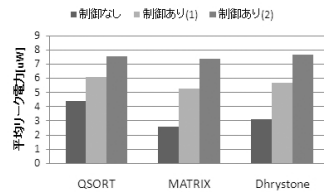


図 10 SHIFT ユニットのスリープ時平均リーク電力

MULT ユニットではタスクが Dhrystone の場合のみリーク電力の増大が見られ，その他の場合では変化がなかった．また DIV ユニットでもリーク電力の変化は見られなかった．スリープ時平均リーク電力の結果からも，BEP の評価に対して考察したように各演算器

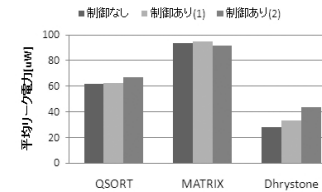


図 11 MULT ユニットのスリープ時平均リーク電力

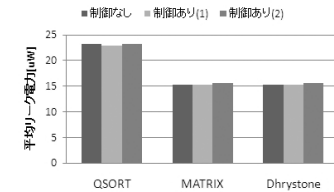


図 12 DIV ユニットのスリープ時平均リーク電力

のスリープ頻度の特性が大きく影響していることがわかる．使用頻度が高く短いスリープが頻発する ALU では制御の効果が現れた一方，SHIFT ユニットでは動的 PG ポリシーのままの方が平均リーク電力が小さいことから，BEP を超えるスリープが多いものと考えられる．

次に，図 13 に評価ケースごとに演算ユニットのスリープ時平均リーク電力を合計し，演算ユニット全体のそれを表したグラフを示す．SHIFT ユニットでは PG ポリシー制御により平均リーク電力が増えているが，一方で ALU での PG ポリシー制御による削減量が大きく，演算ユニット全体としては平均リーク電力を抑えることに成功している．各ベンチマークプログラムの平均において，制御なしの場合に比べて制御あり (1) では約 3%，制御あり (2) では約 10% の平均リーク電力削減を達成している．

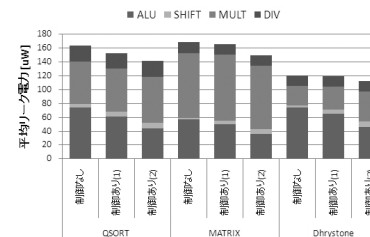


図 13 評価ケースごとの演算ユニット全体のリーク電力

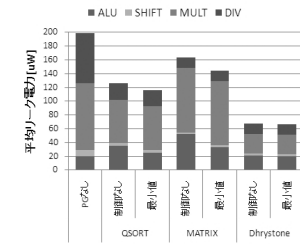


図 14 リーク電力が最小となる制御の組み合わせと制御なし (動的 PG 固定)，PG なしとの比較

続いて，図 14 に演算ユニットごとに最適な制御法を組み合わせさせた場合の電力評価結果を示す．組み合わせは，表 4 のとおりである．このグラフでは，アクティブ時の電力を考慮している．アクティブ時の電力は，BEP の長さだけスリープしたときの平均リーク電力と

トレードオフ関係にあることを利用して温度に応じて試算した値である。また図 13 の一番左の帯は、細粒度パワーゲーティングを行わなかった場合、即ち常にアクティブであった場合の平均リーク電力を示す。平均リーク電力を最小とする組み合わせには、細粒度パワーゲーティングを行わない場合に比べて平均およそ 45%の電力を削減した。また細粒度パワーゲーティングを行いながらも制御を行わなかった場合と比べても、平均約 7%、最大で約 11%の電力を削減した。

細粒度パワーゲーティングを行った場合、使用頻度の低い MULT ユニットや DIV ユニットなどが長くスリープするため、全体では電力が減少ものの、ALU では平均リーク電力は増加する傾向にある。これに対し PG ポリシーの制御を適用することで、BEP により増加した電力を抑制し、細粒度パワーゲーティングによる省電力効果を高めることができる。

表 4 平均リーク電力を最小とする制御法の組み合わせ

ALU	SHIFT	MULT	DIV
制御あり (2)	制御なし	制御なし	制御あり (1)

7.3 パフォーマンスオーバーヘッド

PG ポリシー制御による OS およびタスクを含めた全体に対するパフォーマンスのオーバーヘッドを測定した結果を表 5 に示す。提案方式は平均で約 0.25%程度のオーバーヘッドにとどまっており、小さいオーバーヘッドで省電力効果を得られたといえる。

表 5 PG ポリシー制御のオーバーヘッド

タスク	制御なし [cycle]	制御あり [cycle]	オーバーヘッド [%]
QSORT	2035960	2041546	0.27
MATRIX	2032910	2037530	0.23
Dhrystone	2039498	2034653	0.24

8. おわりに

本稿では、コア温度に対して、BEP の変化を考慮して閾値を設定し Geyser の PG ポリシーを制御する方式を提案した。この方式について、実際に OS に実装し、ベンチマークプログラムをタスクとして実行して評価を行った結果、1 つの閾値に対するコア温度の高低で PG ポリシーを決定する方式で、演算器全体の平均リーク電力を約最大約 11%削減した。

今後「キャッシュミス時のみスリープ」のポリシーよりも「動的 PG」ポリシーの省電力効果が上回るようなケースを見極めることで、OS による動的制御の特徴を生かし更なる電力削減を図る。またテストケースとして、より多様な温度変化、特に周期的に変動する場合などを評価するとともに、スリープ特性の異なるベンチマークプログラムをタスクとして実行することで、提案方式の有効性をさらに検証する。

謝辞 本研究は東京大学大規模集積システム設計教育研究センター (VDEC) を通じ、株式会社半導体理工学研究センター、富士通株式会社、松下電器産業株式会社、NEC エレクトロニクス株式会社、株式会社ルネサステクノロジ、株式会社東芝の協力で行われたものである。

本研究は、科学技術振興機構「JST」の戦略的創造研究推進事業「CREST」における研究領域「情報システムの超低消費電力化を目指した技術革新と統合化技術」の研究課題「革新的電源制御による次世代超低電力高性能システム LSI の研究」による。

参 考 文 献

- 1) 中村宏, 天野英晴, 宇佐美公良, 並木美太郎, 今井雅, 近藤正章, 「革新的電源制御による超低消費電力高性能システム LSI の構想」, 情報処理学会研究報告 ARC-173, pp.79-84, Jun 2007.
- 2) James Donald, Margaret Martonosi, 'Power Efficiency for Variation-Tolerant Multicore Processors', International Symposium on Low Power Electronics and Design (Proc. ISLPED'06) SESSION10, No.3, pp.304-309, Oct 2006.
- 3) Pratap Ramamurthy et al., 'Performance-directed Energy Management using BOS', ACM SIGOPS Operating Systems Review, Vol.41, pp.66-77, 2007.
- 4) Bernhard Egger, Jaejin Lee, Heonshik Shinl, 'Scratchpad Memory Management for Portable Systems with a Memory Management Unit', EMSOFT '06: Proceedings of the 6th ACM & IEEE International conference on Embedded software, pp321-330, 2006.
- 5) 関直臣, Lei Zhao, 徐慧, 池淵大輔, 小島悠, 長谷川揚平, 天野英晴, 香嶋俊裕, 武田清大, 白井利明, 中田光貴, 宇佐美公良, 砂田徹也, 金井遵, 並木美太郎, 近藤正章, 中村宏, 「MIPS R3000 プロセッサにおける細粒度動的スリープ制御の実装と評価」, 情報処理学会研究報告 2008-ARC-176, pp.71-76, Jan 2008.
- 6) 砂田徹也, 関直臣, 香嶋俊裕, 中田光貴, 近藤正章, 天野英晴, 並木美太郎, 「省電力 MIPS プロセッサにおける OS の試作とシミュレーションによる電力評価」, 情報処理学会「システムソフトウェアとオペレーティング・システム」第 108 回研究報告, Vol.2008-OS-108, Apr 2008.
- 7) 砂田徹也, 木村一樹, 近藤正章, 天野英晴, 宇佐美公良, 中村宏, 並木美太郎「省電力を目的とした OS による細粒度パワーゲーティング制御方式の研究」, 情報処理学会「システムソフトウェアとオペレーティング・システム」第 114 回研究報告, Vol.2010-OS-114, Mar 2010.